Visual Simulation of Dust Accumulation

S. C. Hsu *T. T. Wong*
Department of Computer Science
The Chinese University of Hong Kong

Abstract

This paper describes a dust modeling technique. An empirical method is used to visually simulate the effect of dust accumulation on object surfaces. The dust amount is first predicted based on the properties of the surfaces: the surface inclination and stickiness. This predicted amount is then adjusted according to some external factors: surface exposure to wind and scraping off by other objects. The calculated dust amount is finally perturbed by a noise function on rendering to give a fuzzy visual effect.

CR Categories and Subject Descriptors: I.3.3 [Computer Graphics]: Picture/Image Generation - display algorithms; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism - Color, shading and texture.

Additional Key Words: Dust modeling, dust mapping, surface exposure to air.

1 Introduction

Dust accumulation is a common natural phenomenon in real life. Modeling of dust accumulation, however, has not been well studied. Blinn[1] modeled the appearance of dusty surfaces given the thickness of the dust layer. However, he did not provide a method to automatically determine this amount of dust on surfaces. We are presenting a technique in this paper to model the amount of dust settled on object surfaces based on properties of surfaces and the geometry of objects. The scattering of light by air-borne dust has been well-studied[1,8-10] and we shall not consider it in this paper. We shall illustrate our techniques with images of dusty objects.

The determination of the amount of dust accumulated is composed of two phases. In the first phase, we determine a 'normal' amount of dust accumulated on a surface based on properties of the surface: its inclination and stickiness. In the second phase, external factors that affect the normal dust amount will be taken into account and the value is modified. External factors include surface exposure and the scraping off of settled dust by other objects.

The dust accumulation pattern can be stored as a texture map for subsequent use.

2 Dust Amount Prediction Functions

Due to the effect of gravity, horizontal surfaces usually accumulate more dust than inclined ones. However, if an object is placed near an open window in front of a dusty construction site, surfaces facing the window would have more dust accumulated on it than other faces. Therefore inclination of a surface against the direction of the dust source (which might be in terms of the combined effect of gravity, wind direction and the actual source of dust particles) would certainly affect the amount of dust settled. On the other hand, this amount is also dependent on the surface type. A sticky surface (a furry, rough or even adhesive one) is more likely to accumulate dust than less-sticky ones (e.g. a smooth one).

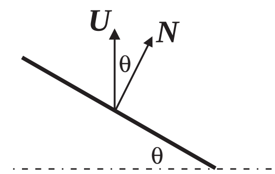

Figure 1: A small surface patch.

Consider a small surface patch with a surface normal N (Figure 1). U is the resultant direction of the dust source. Assuming surface properties are isotropic, the amount of dust accumulated would be a function of θ, the angle between N and U, and s, a stickiness factor of the surface. We shall call this function the *normal dust amount function*. Instead of providing a physical model to the dust adhering process, we shall take the liberty to adopt some functions that approximate the physical process. The function should have its peak when θ is 0 and gradually falls off as θ increases. In real life, even when a surface is upside down, the surface would somehow still have a certain amount of dust adhered on it. Furthermore, an increase in stickiness would decrease the rate of the fall off. These effects are all very similar to the specular reflection of light from a surface. The dust source is analogous to the light source. The accumulation of dust on the wrong side is similar to the diffuse inter-object reflection or an ambient light; the fall off in dust amount with the angle made with the dust source is similar to the fall off in specular high light. Therefore it is natural to consider a function similar to Phong's specular reflection model to model this normal dust amount function:

$$D_U = K/s + (1 - K/s)\cos^s\theta$$

where θ is angle between vector N and U
s is the surface slippiness (large s means less sticky)
K is a constant factor modeling the general adhesion of dust independent of the orientation of the surface.

D_U has all the required properties (Figure 2): A highly sticky surface (solid curve in Figure 2) accumulates larger total dust amount (area under the curve) than a less sticky one (dashed curve in Figure 2). The computation is also efficient, $\cos(\theta)$ is simply the dot product $N \cdot U$.

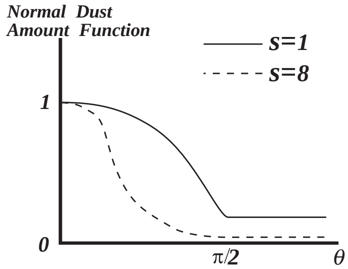

Figure 2: Normal dust amount functions with different values of s. K is 0.2 in both functions.

3 External Factors

To model the influence by external factors like coverage, scraping or wind, the normal dust amount is multiplied by a factor α. α tells us how much the normal dust accumulation amount is realized in the actual environment.

Figure 5 and 6 show a dusty sphere, with a directional dust source from the upper side, generated by multiplying the normal dust amount function by different α.

The value of α is dependent on the geometry of the object and can be further influenced by other factors like wind and scraping.

3.1 Surface Exposure

Consider the object in Figure 3. Area B is in 'shadow' (or more precisely, less exposed to air) while area A is more exposed. As area B has less exposure than area A, the dust amount that area B may *receive* is less. On the other hand, area B may eventually *result* in more dust accumulation — since area B has less exposure, the dust accumulated on it has less chance of being removed by wind or cleared by scraping. Hence the dust amount is more than that on area A.

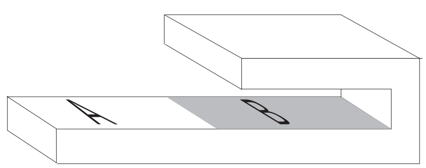

Figure 3: Object with a portion in 'shadow'.

In order to model both cases above, the term α can be written as a function of surface exposure ξ.

$$\alpha = \begin{cases} 0, & \alpha' < 0 \\ \alpha', & 0 \le \alpha' \le 1 \\ 1, & \alpha' > 1 \end{cases}$$

where $\alpha' = \alpha_o + r\xi$
and α_o is a factor modeling the global external effects, $0 \le \alpha_o \le 1$.
ξ is the surface exposure, such that $0 \le \xi \le 1$. Larger ξ means more exposed.
r is a scaling factor to scale the effect of surface exposure. It can be any real number.

When r is positive, the surface exposure is positively related to α. This models the case that area B accumulates less dust than area A. Figure 7 shows one example with positive surface exposure effect. When r is negative, the surface exposure is negatively related to α. This models the opposite case. Figure 8 shows the same object with negative exposure effect.

Figure 4 illustrates how to determine the surface exposure. A number of filler rays are fired from a point P on the surface in random directions in the upper hemisphere to determine the nearest intersection point with any object.

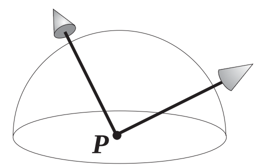

Figure 4: Emitting random rays to detect surface exposure.

The distance d_i from the base point P to the nearest intersecting point of the i-th ray in direction V_i would have a value between 0 and +infinity (when the ray has no intersection). The surface exposure ξ is then defined as:

$$\xi = 1 - \left[\sum_{i=1}^{n} \frac{d_h}{d_h + d_i} \right] \Big/ n$$

where n is number of emitted random rays.
d_i is the distance between P and i-th ray intersection point.
d_h is an user defined constant called the half-exposure distance, which is roughly the average spacing between the object and other nearby objects that would reduce the exposure of the object to 0.5.

In practice, only a few random rays are enough to generate a satisfactory image. Figure 9 and 10 are generated using 1 and 5 random rays respectively. The value of surface exposure would have a greater variation, if less rays are emitted. This would also result in a more scattered appearance.

3.2 Dust Map

Effects due to scraping are modelled using a dust mapping technique, which is a variant of texture mapping[2,3]. The surface dust amount is changed according to the pattern in the dust map, which is just a pixmap. The equation of α is now:

$$\alpha = \begin{cases} 0, & \alpha' < 0 \\ \alpha', & 0 \le \alpha' \le 1 \\ 1, & \alpha' > 1 \end{cases}$$

where $\alpha' = \alpha_o + r\xi + (b_l + \rho(b_u - b_l))$
and ρ is the image pixel value, s.t. $0 \le \rho \le 1$.
$[b_u, b_l]$ is the interval where ρ is mapped to. it is

used to control the perturbation effect of ρ.

Figure 11 shows the dust pattern on a sphere modified with a dust map (on the left in Figure 11).

4 Multiple Dust Sources

Up to now, we have assumed there is only one dust source from a direction U. In some cases, it is more realistic to have multiple dust sources, each from a different direction U_i. For example, an object in a room with two windows opening to a dusty environment. Each dust source can be a directional source (with U_i kept constant), or a point source (with U_i being the vector from a point on the object to the point dust source. The normal dust amount at a particular point on a surface can be further modified by an attenuation factor, which is inversely related to the distance from the dust source). The effect of having multiple dust sources is like multi-pass spraying of dust onto an object (Figure 12). In actual implementation, multi-pass calculation is not necessary however. The final dust amount D is just,

$$D = \min(1, \alpha(D_{U_1} + D_{U_2} + D_{U_3} + \ldots))$$

where D_{U_i} is the normal dust amount function for the i-th dust source.

5 Rendering and Implementation

To render a surface after the amount of dust accumulation has been determined, it is possible to apply the dust amount as the thickness parameter to Blinn's light reflection function[1]. A simpler approach, however, is adopted in our implementation. The final dust amount D is further perturbed by multiplying it with the Perlin's noise function[4]. The final surface properties for rendering are then linearly interpolated with the perturbed dust value between the surface properties when fully covered with dust and the original object surface properties. The surface properties when fully covered with dust are predefined. This linear interpolation of surface properties is just a simplified approximation. A more accurate approach is to precompute an array of BRDF tables[5-7]. Each different table records the BRDF of the surface covered with different amount of dust. On rendering, the actual reflectance can be interpolated among these BRDF tables using the final dust amount.

6 Conclusion

The use of traditional texture mapping technique[2,3] to create the appearance of dusty surfaces requires a lot of manual adjustment and is more difficult to apply to irregularly shaped objects. Furthermore, the distribution of dust on different parts of an object cannot be automatically deduced.

The dust modeling technique described in this paper introduces only a few parameters to control the effect of dust accumulation. The technique automatically determines the dust distribution based on object geometry and control parameters. The stochastic rendering method is simple to implement and resultant images are realistic.

Although we have only applied the technique to model dust accumulation, they can also be applied to other precipitation phenomena. Currently we have assumed the precipitation to have negligible volume. The next step we want to take would be to model precipitations with volume and apply the technique to model other natural precipitations.

References

[1] James F. Blinn, "Light Reflection Functions for Simulation of Clouds and Dusty Surfaces", Computer Graphics (SIGGRAPH '82), Vol. 16, July 1982, pp.21–29.

[2] J. F. Blinn and M. E. Newell, "Texture and Reflection in Computer Generated Images", Communications of the ACM, Vol. 19, No. 10, October 1976, pp. 542–547.

[3] P. S. Heckbert, "Survey of Texture Mapping", IEEE Computer Graphics and Applications, Vol. 6, No.11, November 1986, pp. 56–57.

[4] Ken Perlin, "An Image Synthesizer", Computer Graphics (SIGGRAPH'85), Vol. 19, July 1985, pp. 287–296.

[5] J. T. Kajiya, "Anisotropic Reflection Models", Computer Graphics (SIGGRAPH'85), Vol. 19, July 1985, pp.15–22.

[6] S. H. Westin, J. R. Arvo and K. E. Torrance, "Predicting Reflectance Functions from Complex Surfaces", Computer Graphics (SIGGRAPH'92), Vol. 26, July 1992, pp.255–264.

[7] G. J. Ward, "Measuring and Modeling Anisotropic Reflection", Computer Graphics (SIGGRAPH'92), Vol. 26, July 1992, pp.265–272.

[8] N. Max, "Atmospheric Illumination and Shadows", Computer Graphics (SIGGRAPH'86), Vol. 20, August 1986, pp.117–124.

[9] T. Nishita, Y. Miyawaki and E. Nakamae, "A Shading Model for Atmospheric Scattering Considering Luminous Intensity Distribution of Light Sources", Computer Graphics (SIGGRAPH'87), Vol. 21, July 1987, pp.303–310.

[10] H. E. Rushmeier and K. E. Torrance, "The Zonal Method for Calculating Light Intensities in the Presence of a Participating Medium", Computer Graphics (SIGGRAPH'87), Vol. 21, July 1987, pp.293–302.

Biographical Sketches

Siu-chi Hsu received a B.Sc. in Engineering from the University of Hong Kong with 1st Class Honours in 1986. He continued to do M.Phil. research in Robotics in the same university. He then took a diploma course in computer science in the Cambridge University Computer Lab and graduated with a distinction in 1988. On receiving a few scholarships, he continued his research in Computer Graphics in the Computer Lab and finished his Ph.D. thesis at the end of 1991. He has been a lecturer in the Computer Science Department of the Chinese University of Hong Kong since 1992. He is also one of the founders of Creature House Inc.

Tien-tsin Wong is an M.Phil. student in the Computer Science Department of the Chinese University of Hong Kong. He received his B.S. degree in computer science from the Chinese University of Hong Kong in 1992. He is a member of ACM and SIGGRAPH.

Sidebar: Implementing the Dust Accumulation Technique in a Ray Tracer

Rayshade ver. 4.0.6 is a fairly popular public domain ray tracer written by Craig Klob et. al. We have implemented the technique as a texture module in Rayshade. The descriptions of objects, light sources, surface attributes and other textures are all independent of the dust module. To apply some dust to any existing scenes involves simply adding a few lines to the existing 4 description file. The user just have to introduce a few parameters to the surface attribute list of the surfaces to be affected, and describe where are the dust sources.

The calculation of dust amount is performed on the fly during the rendering process. Only the visible positions on the surfaces in a scene will have the dust amount calculated. Therefore it is quite efficient. This also implies that on changing the view point, all the dust amount calculations would have to be performed again. It is however possible to have the dust amount calculations done in a preprocessing pass for every surface and record the results in texture maps. This would certainly improve the performance of rendering animation sequences. Such a preprocessing pass would also be necessary if the technique is to be used in Z-buffer based rendering systems.

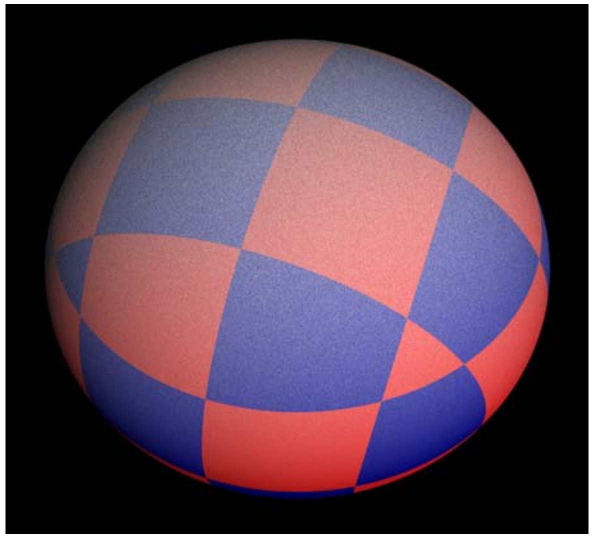

Figure 5: A dusty sphere (top view). α=1, s=1 and K=0.

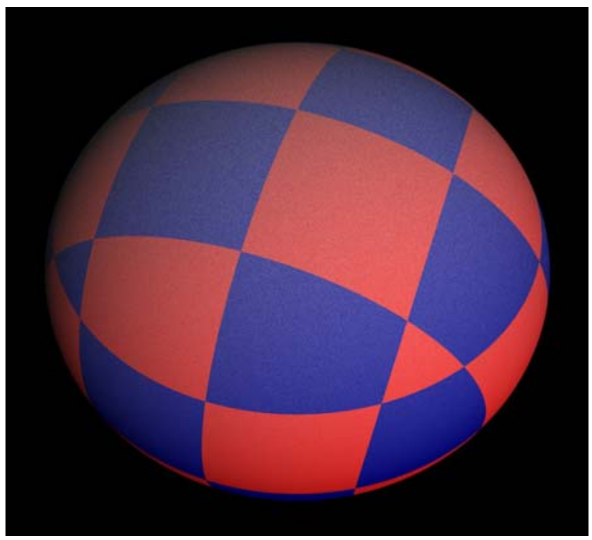

Figure 6: The same dusty sphere. α=0.6, s=1 and K=0.

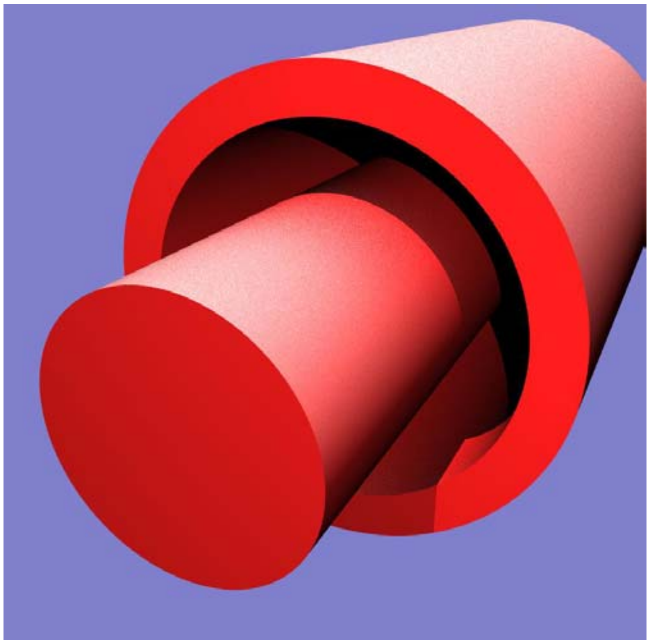

Figure 7: The effect of using a positive exposure scaling factor: less-exposed area received less dust.

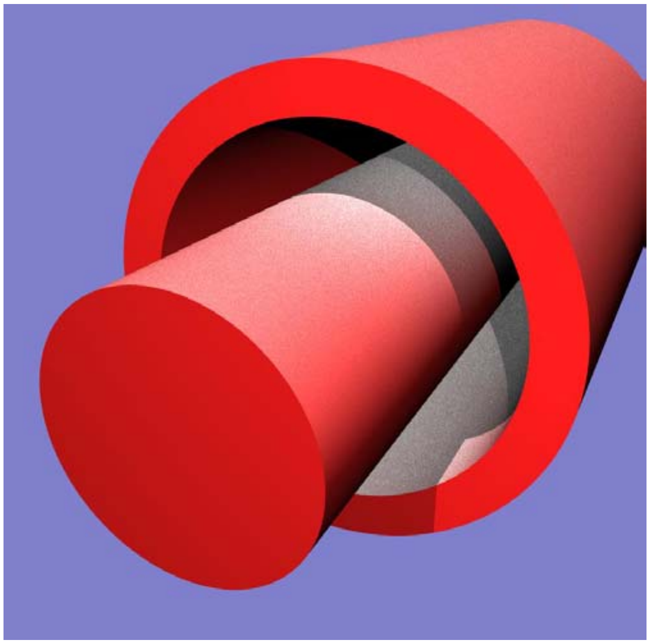

Figure 8: The effect of using a negative exposure scaling factor: less-exposed area accumulated more dust.

Figure 9: A dusty door-plate with the word “Dust!” on it. One random ray is casted per pixel in determining the surface exposure.

Figure 10: The same door-plate. Five random rays are casted per pixel to determine the surface exposure.

Figure 11: The previous dusty sphere modified with the dust map on the left.

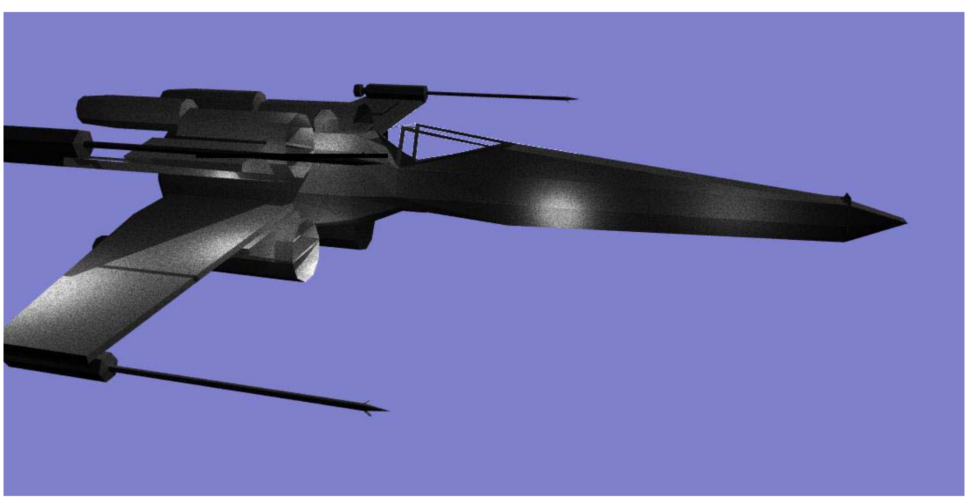

Figure 12: A dusty X-wing fighter ‘sprayed’ with 6 point dust sources.